

# **Uživatelská dokumentace k nástroji jSQLBenchmark 1.0**

**Ondřej Čabrada**

**Pavel Král**

© **Copyright 2007-2008** Katedra informatiky a výpočetní techniky Západočeské univerzity v Plzni.

Tato aplikace je poskytována pod licencí Creative Commons Attribution Non-commercial Share Alike, další informace na <http://creativecommons.org/licenses/by-nc-sa/3.0/>.

## 1. Úvod

Aplikace `jSQLBenchmark 1.0` slouží k jednoduchému testování relačních databázových systémů. Její metrikou je počet dotazů vykonaných za sekundu v závislosti na počtu paralelně přístupujících uživatelů.

Aplikace disponuje grafickým uživatelským rozhraním, což činí práci s programem příjemnější. Veškeré parametry testu a parametry připojení k databázi se nastavují prostřednictvím k tomu určených komponent v uživatelském rozhraní, kde se rovněž zobrazují výsledky testu v grafické či tabulkové podobě.

Výhodou aplikace oproti jiným dostupným benchmarkům je skutečnost, že ji lze použít k otestování libovolné databáze s libovolnou datovou strukturou. To je zajištěno díky tomu, že aplikace nepoužívá žádnou interní (napevno zabudovanou) datovou strukturu, nad kterou se vykonávají předem specifikované testovací dotazy. Při použití `jSQLBenchmarku` se předpokládá, že si uživatel sám určí strukturu databáze, nad kterou bude provádět vlastní sadu testovacích dotazů.

Připojení k testované databázi je realizováno přes JDBC ovladač příslušného databázového systému, což umožňuje benchmark snadno rozšiřovat o podporu jiných databází.

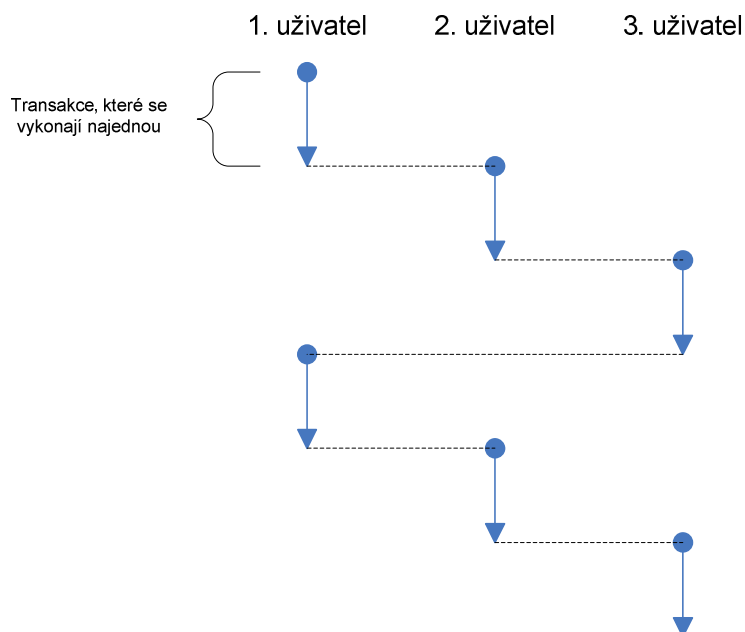
Díky použité technologii (Java 1.6) je aplikace bez jakýchkoliv úprav přenositelná na různé operační systémy podporující Java Virtual Machine. Testována byla pod operačními systémy Microsoft Windows XP SP2, Linux Mandrake 10.5 Limited Edition a Ubuntu 6.10 Ultimate.

## 2. Princip testování

Ještě než si popíšeme uživatelské rozhraní aplikace a její ovládání, řekneme si něco o tom, na jakém principu je založen vlastní algoritmus testu.

Program simuluje virtuální uživatele, kteří pseudoparalelně přistupují k testované databázi a vykonávají nad ní zadanou sadu testovacích transakcí. Transakce se mohou skládat z jednoho nebo více SQL dotazů.

V rámci testu lze nastavit, kolik transakcí bude vykonávat každý virtuální uživatel a kolik z celkového počtu transakcí vykoná najednou, než dojde k přepnutí na jiného uživatele (pseudoparalelní běh jednotlivých uživatelů je znázorněn na obrázku 2.1).



**Obrázek 2.1 Pseudoparalelní běh virtuálních uživatelů**

Přepínání jednotlivých uživatelů nemusí být pravidelné jako na výše uvedeném obrázku, záleží na Java Virtual Machine a operačním systému, jak bude jednotlivé virtuální uživatele vybírat.

Pro každou testovací transakci lze nastavit její procentuální četnost vztahující se k celkovému počtu transakcí, které jeden virtuální uživatel v rámci testu vykoná. Pokud tedy budou zadány tři testovací transakce, kde první bude mít četnost 40%, druhá 50% a třetí 10% a celkově vykoná uživatel 10 transakcí, znamená to, že první dotaz bude zastoupen čtyřikrát, druhý dotaz pětkrát a třetí dotaz jednou. Pořadí, ve kterém se jednotlivé transakce vykonávají je řízeno náhodně (randomem) a pro každého virtuálního uživatele může být jiné.

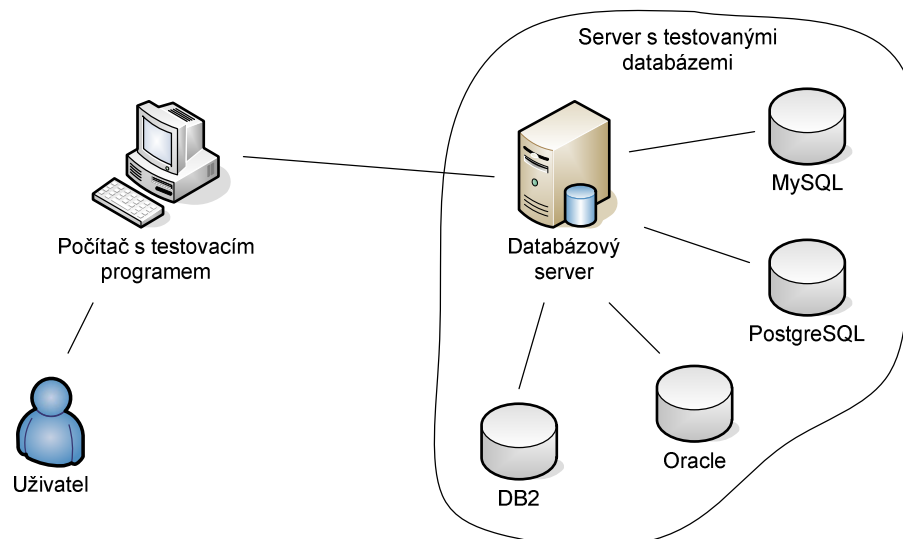
Posledním parametrem, který se v testu nastavuje, je počet „zahřívacích transakcí“. To je počet transakcí, které musí každý virtuální uživatel vykonat ještě před započítáním vlastního ostrého testu. Jejich úkolem je zajistit, aby v okamžiku ostrého testu již byli všichni uživatelé připojeni k databázi. V této fázi ještě měření výkonu neprobíhá a ostrý test se spouští teprve poté, co všichni uživatelé projdou „zahřívacím kolem“ testu.

V režimu ostrého testu se měří doba odezvy na jednotlivé transakce, která se převádí na průměrnou dobu odezvy na jeden dotaz. V grafickém vyjádření se pak zobrazuje závislost počtu dotazů za sekundu na počtu virtuálních uživatelů.

## 2.1. Vytvoření testovacího prostředí

Testovací aplikaci lze spouštět na stejném počítači, jako je nainstalována databáze. Tento způsob ale nelze obecně příliš doporučit, protože při větším počtu uživatelů se může zdát, že dochází k většímu snížení výkonu, než je tomu ve skutečnosti. Důvodem je skutečnost, že testovací aplikace také zabírá nějaké prostředky (procesor, paměť apod.) a když bude probíhat test pro větší počet uživatelů, bude testovací aplikace ubírat více prostředků na úkor testované databáze, což se může projevit citelnějším snížením výkonu.

Z toho důvodu je výhodnější spouštět benchmark na samostatném počítači a testovanou databázi provozovat na samostatném serveru. Komunikaci lze zajistit síťovými prostředky (viz obrázek 2.2).



Obrázek 2.2 Testovací prostředí

Samozřejmě je třeba počítat s určitým zpožděním po komunikačních linkách, ale pokud nebude síť zatížena žádnými jinými přenosy, budou mít všechny testované databáze prakticky stejné podmínky a dosažené výsledky bude možné mezi sebou srovnávat.

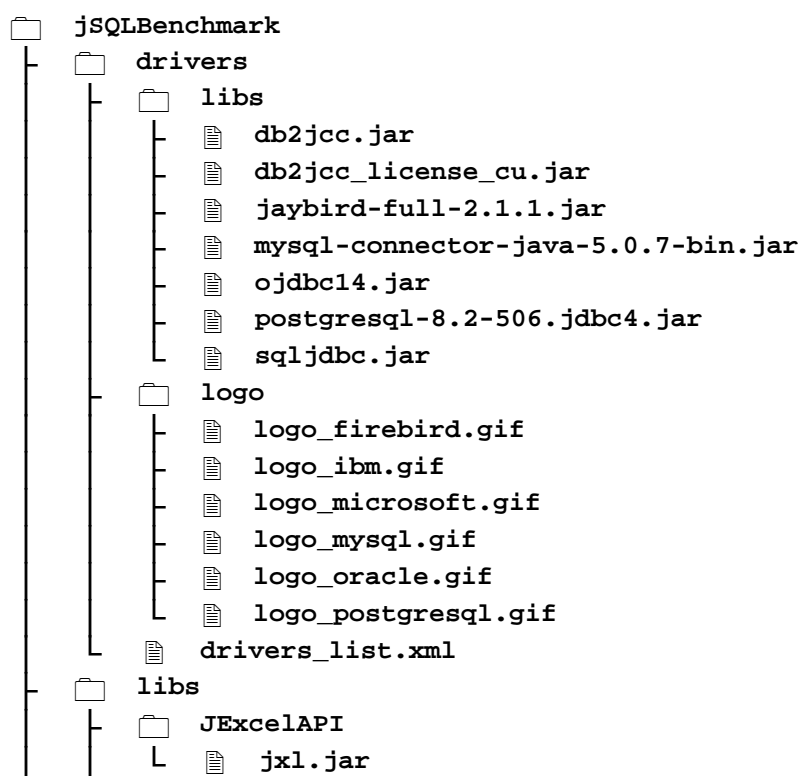


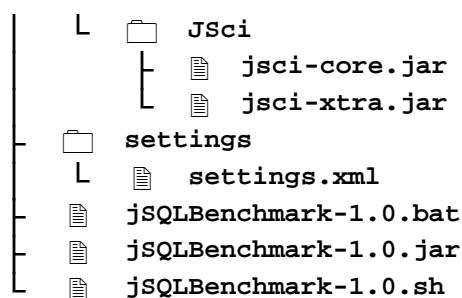
**Tip:** Oddělte testovací nástroj od serveru s databázovými systémy, u nichž provádíte měření, aby nedocházelo k ovlivňování výsledků v důsledku spotřebovávání zdrojů (paměti a času procesoru) samotným benchmarkem.

## 3. Instalace a spuštění aplikace

### 3.1. Instalace a struktura aplikace

Aplikace je distribuována v archívu ZIP a instalace se provádí pouhým rozbalením archívu do zvoleného adresáře. Výsledný produkt má následující strukturu:





### 3.2. Požadavky na program

Pro svou činnost vyžaduje aplikace nainstalované běhové prostředí Javy verze 1.6 (označované také jako verze 6), stačí pouze Java 2 Standard Edition Runtime Environment 6 (JRE), v případě, že používáte Java 2 Standard Edition Development Kit 6 (JDK), je JRE nainstalováno automaticky. Veškeré nástroje pro Javu je možno stáhnout z <http://java.sun.com/>.

### 3.3. Spuštění aplikace

Spuštění aplikace je možné ve Windows z konzole zadáním příkazu (za předpokladu, že je správně nastavena cesta k instalaci Javy v proměnné `JAVA_HOME`):

```
C:\jSQLBenchmark>jSQLBenchmark-1.0.bat
```

nebo z terminálu v Linuxu:

```
user@localhost:~/jSQLBenchmark$ ./jSQLBenchmark-1.0.sh
```

Druhou možností je spuštění aplikace přímo kliknutím na jméno (ikonku) spustitelného souboru `jSQLBenchmark-1.0.bat` resp. `jSQLBenchmark-1.0.sh`.

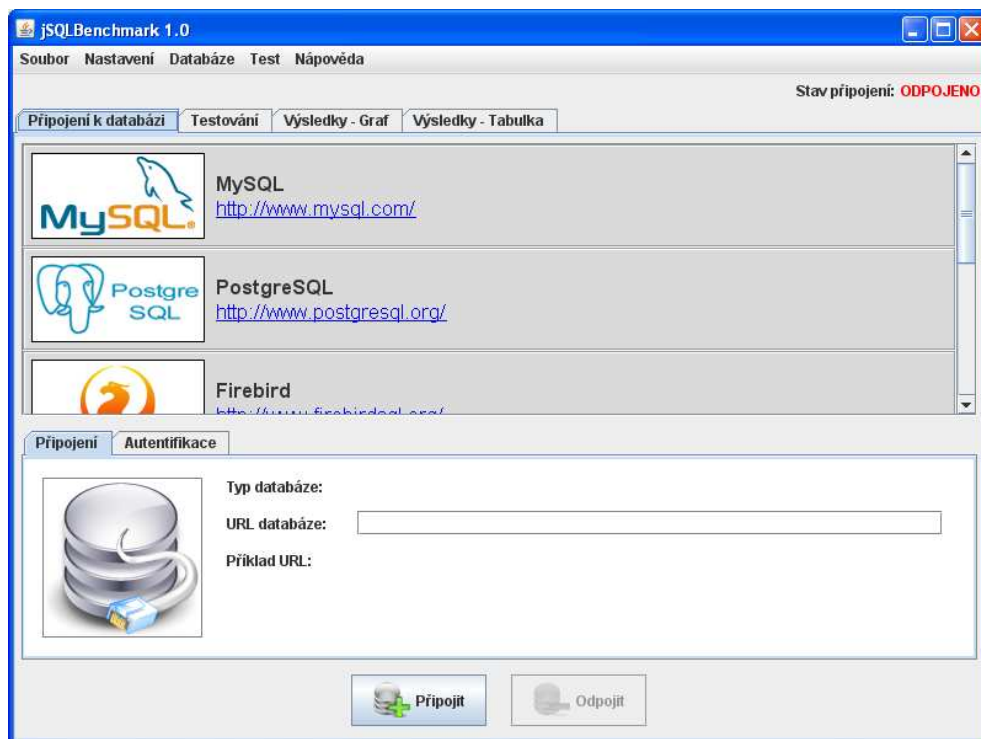


**Upozornění:** *Nespouštějte aplikaci přímým spuštěním archivního souboru `jSQLBenchmark-1.0.jar`. Dojde sice ke spuštění aplikace, ale z důvodu nenačtení databázových ovladačů nebude možné připojit se k žádné z nabízených databází!*

## 4. Seznámení s uživatelským rozhraním

Uživatelské rozhraní aplikace je navrženo intuitivně tak, aby práce s nástrojem byla co nejjednodušší. Na obrázku 3.1 je zobrazeno hlavní okno aplikace po jejím spuštění.

Veškeré hlavní ovládací panely jsou dostupné z hlavního okna aplikace přes záložky. Program je založen na filozofii průvodce, kdy jednotlivé záložky (v pořadí za sebou) odpovídají úkonům, které je třeba udělat pro spuštění testu a zobrazení výsledků. Popis panelů pod jednotlivými záložkami je obsahem následující části.



Obr. 3.1 Hlavní okno aplikace po spuštění

## 4.1. Záložky

Záložky jsou celkem čtyři, zobrazené v pořadí úkonů, které je třeba provést od připojení k testované databázi, přes spuštění vlastního testu až po zobrazení výsledků. Nyní se podíváme, co nám jednotlivé karty nabízí. Popis konkrétního ovládání je uveden v kapitole 5.

### 4.1.1. Karta Připojení k databázi

Na kartě „*Připojení k databázi*“ jsou hlavní ovládací prvky pro specifikaci připojení. V horní části se nachází seznam s dostupnými JDBC ovladači. Po výběru některé z nich je možné na kartách v dolní části specifikovat parametry připojení (URL adresu k databázi, přihlašovací jméno a heslo). URL adresa se nastavuje na kartě „*Připojení*“, uživatelské přístupové údaje na kartě „*Autentifikace*“.

Připojení k databázi, či odpojení od databáze se provádí kliknutím na příslušné tlačítko. V odpojeném stavu je dostupné pouze tlačítko „*Připojit*“, v připojeném stavu je dostupné pouze tlačítko „*Odpojit*“ (aby se zabránilo připojování k více databázím najednou).

### 4.1.2. Karta Testování

Karta „*Testování*“ je určena pro specifikování parametrů benchmarkového testu. Je přehledně rozdělena na tři části. V levém horním panelu se specifikují parametry týkající se virtuálních uživatelů, v pravém horním panelu se zadávají vlastnosti testovacích transakcí a ve spodním panelu se definuje sada vlastních testovacích transakcí. Transakce je možno přidávat, mazat a editovat prostřednictvím tlačítek umístěných pod tabulkou dotazů. Pro spuštění testu slouží tlačítko „*Spustit test*“.

### 4.1.3. Karta Výsledky – Graf a Výsledky – Tabulka

Poslední dvě karty „*Výsledky – Graf*“ a „*Výsledky – Tabulka*“ slouží pro prezentaci výsledků testu v grafické či tabulkové podobě. Každé z těchto zobrazení má své výhody – grafické je přehlednější, tabulkové podrobnější a přesnější.

## 4.2. Menu

Kromě záložek disponuje aplikace rovněž ovládacím menu, které poskytuje pět různých kategorií. Některé funkce menu jsou provázané s ovládacími panely dostupnými přes záložky (viz dále).

### 4.2.1. Menu Soubor

Menu „*Soubor*“ se převážně vztahuje na záložky „*Výsledky – Graf*“ a „*Výsledky – Tabulka*“. Umožňuje uložení výsledků do souboru, načtení výsledků ze souboru (pro opětovnou prezentaci), export tabulky výsledků do Excelu a export tabulky výsledků a grafu do grafických souborů (podporovány jsou formáty gif, jpeg, png a bmp).

Kromě výše uvedeného je možné přes menu „*Soubor*“ aplikaci ukončit.

### 4.2.2. Menu Nastavení

Menu „*Nastavení*“ se týká uživatelského nastavení prostředí aplikace. Je možno zvolit jeden ze dvou předdefinovaných vzhledů a jazyk aplikace (v základní verzi je dostupná čeština a angličtina). Uživatelské nastavení lze přes menu „*Soubor*“ rovněž uložit.

### 4.2.3. Menu Databáze

Menu „*Databáze*“ se vztahuje ke kartě „*Připojení k databázi*“. Kromě samotného připojení a odpojení databáze (které je dostupné rovněž na příslušné kartě ve formě tlačítek), umožňuje uložení parametrů databázového připojení do souboru či jejich opětovné načtení ze souboru. Díky tomu je možné se ke stejné databázi připojovat opakovaně, aniž by bylo nutné znovu vyplňovat přístupové údaje.

### 4.2.4. Menu Test

Menu „*Test*“ se vztahuje ke kartě „*Testování*“. Umožňuje spustit test, vytvořit nový test a uložit či načíst parametry testu do/ze souboru. Díky tomu je možné jedno nastavení testu použít opakovaně, bez nutnosti opětovného (často zdlouhavého) vyplňování.

### 4.2.5. Menu Nápověda

V poslední kategorii menu – „*Nápověda*“ lze pouze zobrazit informace o programu. Interaktivní nápověda k programu v současné verzi není k dispozici.

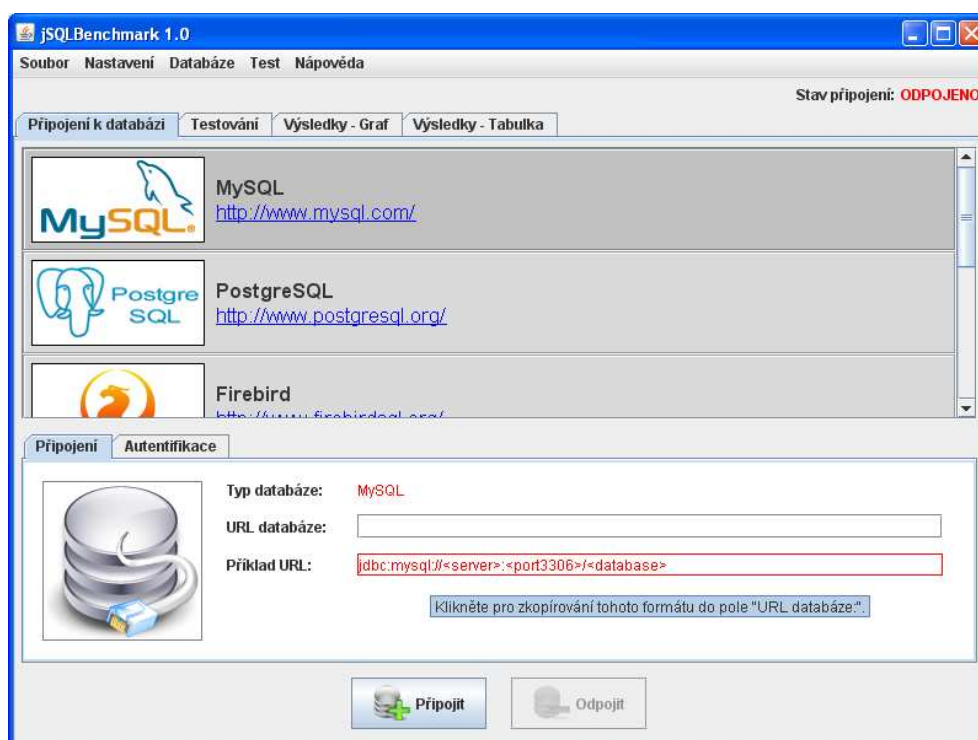
## 5. Ovládání programu

Nyní se podíváme na vlastní ovládání aplikace. Celou funkčnost si předvedeme prakticky na příkladu. Uvedeme si postupně všechny kroky, které je třeba vykonat pro úspěšné provedení testu.

### 5.1. Výběr JDBC ovladače a specifikace připojení

V první řadě je třeba vybrat databázi, kterou budeme testovat (ukážeme si to pro případ databáze MySQL). Na kartě „*Připojení k databázi*“ si v seznamu dostupných JDBC ovladačů

vybereme ovladač MySQL. Po výběru se ve spodním panelu na kartě „**Připojení**“ zobrazí základní informace o vybraném ovladači – jméno a příklad URL (viz obrázek 5.1).



Obr. 5.1 Výběr JDBC ovladače

Nyní je třeba specifikovat URL databáze a na kartě „**Autentifikace**“ ještě přihlašovací jméno a heslo (viz obrázky 5.2 a 5.3).



Obr. 5.2 Specifikace URL řetězce pro připojení k databázi



Obr. 5.3 Zadání přihlašovacích údajů

Nyní je již možné se k databázi připojit. K tomu jsou dvě možnosti – buď kliknutím na tlačítko „**Připojit**“, nebo přes menu „**Databáze – Připojit**“. V případě úspěšného připojení se v hlavním okně aplikace v pravém horním rohu změní stav připojení z **ODPOJENO** na **PŘIPOJENO**. V případě neúspěchu se zobrazí dialogové okno s popisem chyby, ke které došlo (nejčastější příčinou jsou chybně zadané přihlašovací údaje).



Pokud si chcete parametry připojení k databázi uložit pro opakované použití, vyberte v menu „*Databáze*“ položku „*Uložit připojení*“. V případě, že již nějaké připojení máte uloženo, je jeho načtení možné přes menu „*Databáze – Načíst připojení*“. Vzhledem k tomu, že se při načtení parametrů připojení automaticky vybírá JDBC ovladač, je nutné, aby v seznamu ovladačů existoval. V opačném případě načtení připojení neproběhne a zobrazí se chybová zpráva.

## 5.2. Definování parametrů testu

Pokud připojení k databázi proběhlo úspěšně, můžeme přejít k dalšímu kroku, a to ke specifikování parametrů benchmarkového testu. Přejdeme na kartu „*Testování*“. Na panelu „*Počet virtuálních uživatelů (klientů)*“ se zadává, pro jaké počty uživatelů se bude test provádět. Dolní a horní mez udávají minimální a maximální počet virtuálních uživatelů a krok specifikuje, o kolik se zvýší počet uživatelů pro následující test. Pro lepší pochopení si to ukážeme na příkladu. Budeme chtít provést test pro 5, 10, 15, 20, 25 a 35 virtuálních uživatelů. V takovém případě zvolíme dolní mez 5, horní mez 35 a krok 5. Pro případ, že budeme test realizovat pro 1 až 7 uživatelů s krokem 1, bude nastavení vypadat jako na obrázku 5.4.

Obr 5.4 Nastavení parametrů virtuálních uživatelů

Na panelu „*Vlastnosti transakcí*“ se nastavuje jednak celkový počet transakcí, které má jeden virtuální uživatel provést, dále počet transakcí, které vykoná najednou a počet transakcí, které budou provedeny v rámci zahřívacího testu. Opět si to vysvětlíme na příkladu. Jednotliví virtuální uživatelé nepřistupují k databázi skutečně paralelně, ale pseudoparalelně. V případě nastavení jako na obrázku 5.5 bude jeden uživatel vykonávat 20 transakcí (v jakém přesném složení bude popsáno dále). Uvažujme dále test, kdy do databáze střídavě přistupuje 5 virtuálních uživatelů. Pak uživatel, který se připojí nejrychleji, vykoná 4 transakce, následně se přepne na jiného uživatele, který opět vykoná 4 transakce. V okamžiku, kdy se řízení vrátí k prvnímu uživateli, vykoná tento další 4 transakce. Tímto způsobem probíhá test, dokud všichni uživatelé nevykonají předepsaných 20 transakcí. Nutno ještě poznamenat, že přepínání virtuálních uživatelů je náhodné. Ještě než se spustí ostré testování se zaznamenáváním doby odezvy databázového systému, každý uživatel provede 5 transakcí bez měření v rámci zahřívacího kola (aby v okamžiku ostrého testu měla databáze vytvořena sezení („sessions“) pro všechny virtuální uživatele).

Obr 5.5 Nastavení vlastností testovacích dotazů

Poslední věc, kterou je třeba specifikovat, je sada testovacích transakcí. Testovací transakce se přidávají do tabulky přes dialogové okno vyvolané tlačítkem „*Přidat transakci*“ (viz obrázek 5.6). V dialogovém okně se zadávají dvě položky – samotné SQL dotazy tvořící transakci (každý

dotaz musí být ukončen středníkem!) a jejich relativní četnost (v procentech) vůči celkovému počtu dotazů vykonaných jedním uživatelem.

Obr. 5.6 Přidání SQL dotazu

Transakce je možné smazat nebo editovat (k tomu slouží tlačítka „*Smazat transakci*“ a „*Editovat transakci*“). Před smazáním či editací je třeba v tabulce nejprve označit řádku s transakcí, které se to týká.

Kompletní sada testovacích transakcí může vypadat jako na obrázku 5.7. Pro výše uvedené nastavení (každý uživatel vykonává celkem 20 dotazů) to znamená, že první SELECT se vykoná jednou, druhý SELECT se vykoná osmkrát a třetí SELECT se vykoná jedenáctkrát (je to postupně 5% z 20, 40% z 20 a 55% z 20).

SQL transakce	
SQL transakce	Vyvážení [%]
SELECT * FROM oddeleni;	5
SELECT * FROM oddeleni WHERE nazev = 'KIV';	40
SELECT * FROM oddeleni WHERE nazev = 'CIV';	55

Přidat transakci
 Smazat transakci
 Editovat transakci

Obr. 5.7 Specifikace sady testovacích dotazů

Jedna transakce se nemusí skládat pouze z jednoho dotazu jako na obrázku 5.7, ale je možno zadávat více dotazů oddělených středníky. To se dá s výhodou využít v případě, kdy potřebujeme, aby se některé dotazy vykonaly atomicky v přesně definovaném pořadí. Například v internetovém obchodě bude objednávka realizována zápisem do tabulky s objednanými položkami, kam se zapíše vybrané exempláře, a dále zápisem do tabulky s objednávkami, kde se zaznamená číslo objednávky, datum a celková cena. V takovém případě se zapíše všechny potřebné dotazy do jedné transakce. **Pozor: Nikdy nezapisujte dotazy, u kterých požadujete vykonání v přesně definovaném pořadí nebo aby se vykonaly všechny najednou, jako samostatné transakce. Vykonávání transakcí je řízené zcela náhodně a na základě nastavené četnosti!**

Pro zvýšení atraktivnosti testu je možné do SQL dotazů vkládat proměnné. Proměnné jsou definovány čtyři, a sice VIRTUAL-USER, STRING, NUMBER a COUNTER. Proměnné se přímo vkládají do SQL dotazů a uzavírají se složenými závorkami. V tabulce 5.1 jsou jednotlivé proměnné vysvětleny, v tabulce 5.2 je ukázáno jejich použití.

Proměnná	Význam
VIRTUAL-USER	Proměnná obsahuje ID (identifikační čísla) jednotlivých virtuálních uživatelů. Pokud bude test prováděn pro 5 uživatelů, bude mít první uživatel ID = 1, druhý ID = 2 až pátý ID = 5.
NUMBER	Proměnná obsahuje náhodné číslo v zadaném intervalu.
STRING	Proměnná obsahuje náhodný řetězec (počet znaků je definován zadaným intervalem).
COUNTER	Proměnná definuje globální čítač, který v rámci jednoho testu produkuje čísla od 1 výše s inkrementem 1.

**Tabulka 5.1 Typy proměnných**

Příklad proměnné	Vysvětlení funkčnosti
{VIRTUAL-USER}	Proměnnou lze využít, pokud potřebujeme v dotazu brát v úvahu uživatele, který jej provádí. Například budeme mít tabulku se zákazníky internetového obchodu, které lze jednoznačně identifikovat podle hodnoty ve sloupci id_zakaznika (hodnoty budou 1, 2, 3, ... počet zákazníků). Pak při vykonání dotazu, kterým zákazník zobrazuje přehled svých objednávek, můžeme s výhodou použít právě tuto proměnnou, která v případě virtuálního uživatele s ID = 1 bude zobrazovat objednávky zákazníka s id_zakaznika = 1, v případě virtuálního uživatele s ID = 2 bude zobrazovat objednávky zákazníka s id_zakaznika = 2 atd.
{NUMBER\$FROM=10\$TO=50}	Proměnná umožňuje do dotazu vložit náhodné číslo v rozmezí FROM-TO (pro příklad z vedlejšího sloupce bude proměnná obsahovat náhodná čísla od 10 do 50).
{STRING\$FROM=10\$TO=20}	Proměnná umožňuje do dotazu vložit náhodný řetězec s délkou v rozmezí FROM-TO (pro příklad z vedlejšího sloupce bude proměnná obsahovat náhodné řetězce s délkou od 10 do 20).
{COUNTER}	Produkce v rámci jednoho testu čísla od jedné výše (při dalším testu počítá opět od jedné).

**Tabulka 5.2 Příklady použití proměnných**

Proměnné NUMBER a STRING lze s výhodou použít v příkazech INSERT a UPDATE, kdy nechceme, aby ukládané hodnoty byly pořád stejné. Použití výše popsanych proměnných v dotazu je následující (před vykonáním vlastního SQL dotazu dojde k jejich nahrazení konkrétními hodnotami):

```
SELECT * FROM objednávky WHERE id_zakaznika = {VIRTUAL-USER};

INSERT INTO zakaznik (jmeno, prijmeni, vek)
VALUES(' {STRING$FROM=5$TO=10}', ' {STRING$FROM=5$TO=20}',
      {NUMBER$FROM=18$TO=99});
```

Proměnná COUNTER (čítač) se dá použít všude tam, kde potřebujeme, aby se v rámci testu měnily hodnoty od 1 výše s přírůstkem 1.



**Upozornění:** *V případě chybně zadaných proměnných (chybný formát zápisu, chybějící ohraničující závorky, hodnota FROM větší než hodnota TO apod.) se řetězec s proměnnou nenahrazuje konkrétní hodnotou! To ve většině případů vede na chybnou syntaxi SQL příkazu a na předčasné ukončení testu!*



**Tip:** *Jednotlivé proměnné se dají navzájem kombinovat. Tak je možné vygenerovat například řetězec, ke kterému se připojí číslo {STRING\$FROM=10\$TO=20}{COUNTER} a podobně.*

Veškeré parametry testu je možno stejně jako v případě parametrů pro připojení k databázi uložit do souboru pro opakované použití. Uložení se provádí přes menu „**Test – Uložit test**“, načtení testu přes „**Test – Načíst test**“. Vrácení celého formuláře do výchozího stavu je možno přes menu „**Test – Nový test**“. Tím dojde k nastavení výchozích hodnot parametrů a k vyprázdnění sady testovacích transakcí.

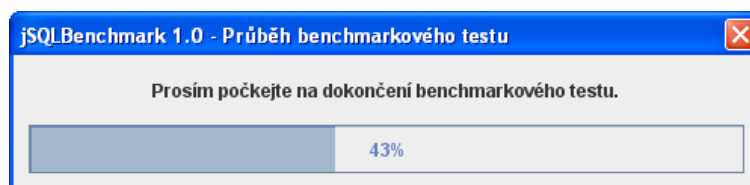
### 5.3. Spuštění testu a prezentace výsledků

Po specifikování všech parametrů testu, je možno test spustit přes tlačítko „**Spustit test**“ nebo přes menu „**Test – Spustit test**“. Test se spustí pouze v případě, že jsou všechny parametry nastaveny korektně. Co to znamená je uvedeno v tabulce 5.3.

Skupina parametrů	Omezení
Počet virtuálních uživatelů	dolní mez < horní mez $0 < \text{krok} \leq (\text{horní mez} - \text{dolní mez})$
Vlastnosti transakcí	počet transakcí vykonaných najednou $\leq$ počet vykonaných transakcí počet zahřívacích transakcí $\leq$ počet vykonaných transakcí
Sada SQL transakcí	součet relativních četností všech transakcí musí být roven 100% v tabulce transakcí musí být alespoň jedna transakce

**Tabulka 5.3 Omezení parametrů testu**

Pokud jsou všechny parametry v pořádku, dojde ke spuštění testu. Při testu se skryje hlavní okno aplikace a zobrazuje se pouze ukazatel průběhu testu (viz obrázek 5.8). Po ukončení testování se hlavní okno opět zobrazí a automaticky se přepne na kartu „**Výsledky - Graf**“.



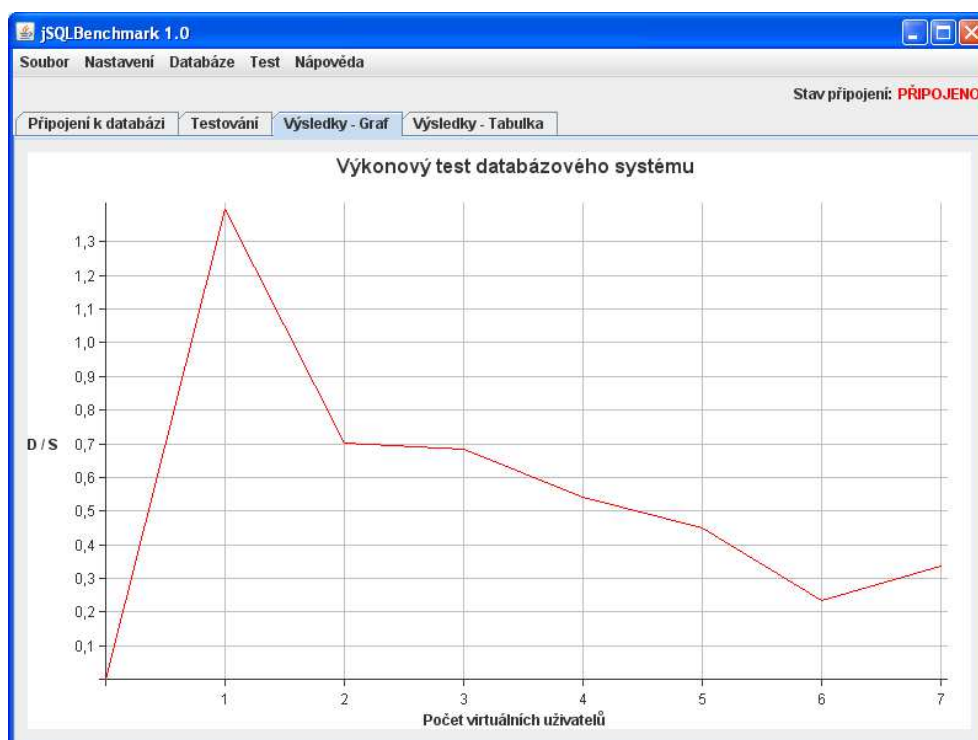
**Obr. 5.8 Ukazatel průběhu testu**



**Upozornění:** *Průběh testu v závislosti na nastavených parametrech, rychlosti databázového serveru, výkonu samotné databáze, velikosti databázových tabulek apod. může trvat od několika minut až po několik hodin!*

## 5.4. Ukládání, prohlížení a export výsledků

Po ukončení testu jsou výsledky zobrazeny na kartách „*Výsledky – Graf*“ (obrázek 5.9) a „*Výsledky – Tabulka*“ (obrázek 5.10). Karta s výsledky v podobě grafu se po dokončení testu zobrazuje automaticky.



Obr. 5.9 Zobrazení výsledků v grafické podobě

Počet uživatelů	Doba odezvy systému [s]	Počet dotazů za sekundu (QpS)
0	0.0	0.0
1	0.71605	1.3965505202150688
2	1.427825	0.7003659412042793
3	1.4622666666666664	0.6838697911917572
4	1.8476625	0.5412243848646601
5	2.22701	0.44903255935087855
6	4.288849999999999	0.23316273593154346
7	2.9581357142857145	0.3380507510763294

Obr. 5.10 Zobrazení výsledků v tabulkové podobě

Výsledky je možno uložit do XML souboru přes menu „*Soubor – Uložit*“. Pokud si chcete prohlédnout některé ze starších výsledků, lze je načíst v menu „*Soubor – Otevřít*“.

Pro možnost dalšího zpracování výsledků disponuje aplikace funkcí „*Export do Excelu*“, která je dostupná rovněž v menu „*Soubor*“.

V neposlední řadě umožňuje aplikace export grafu i tabulky do grafického souboru (na výběr jsou formáty gif, jpg, png a bmp. Funkce je dostupná v menu „*Soubor – Export do grafiky*“, kde je na výběr „*Export tabulky výsledků*“ a „*Export grafu*“.

## 6. Nastavení programu

### 6.1. Základní nastavení uživatelského prostředí

V této části si řekneme něco o možnostech nastavení uživatelského rozhraní aplikace. Možnosti jsou dvě, a to změna vzhledu a změna jazyka. Veškeré nastavení lze uložit (ukládá se do souboru /settings/settings.xml). V případě opětovného spuštění aplikace se načítá automaticky. Pokud soubor s nastavením neexistuje, nastaví se výchozí hodnoty.

#### 6.1.1. Vzhled aplikace

Vzhled aplikace je možno změnit prostřednictvím menu „*Nastavení – Vzhled aplikace*“, kde jsou na výběr dva typy – *Metal* a *Windows*.

#### 6.1.2. Jazyk aplikace a možnosti lokalizace do dalších jazyků

Další možností je změna jazyka aplikace. V současné verzi je k dispozici kompletní lokalizace v českém a anglickém jazyce. Volba jazyka se provádí v menu „*Nastavení – Jazyk*“.

Lokalizační soubory jsou umístěny uvnitř archívu jsQLBenchmark-1.0.jar v adresáři language. Pokud chcete lokalizovat aplikaci do jiného jazyka, vytvořte soubor lang\_xx\_YY.properties resp. lang\_xx.properties podle dostupné šablony. Místo xx a YY zadejte kód jazyka (xx) a případně i kód země (YY). Seznam možných kódů je uveden v souboru lang\_codes.txt, šablona pro překlad je v souboru lang\_template.properties. Nyní stačí přeložit jednotlivé texty z aplikace (překládá se pouze text za znakem =). V níže uvedené ukázce se překládá pouze text označený modře. Identifikátory před znakem = se nesmí měnit!

```
Menu_File=File
Menu_File_Open=Open
Menu_File_Save=Save
Menu_File_Export_XLS=Export to Excel
Menu_File_Export_Graphics=Export to Graphic File
Menu_File_Export_Graphics_Table=Export of Result Table
Menu_File_Export_Graphics_Graph=Export of Graph
Menu_File_Exit=Exit
```

Po přeložení je třeba výsledný soubor přibalit do JAR archívu aplikace do adresáře language.

Při dodržení výše uvedených zásad si aplikace při spuštění automaticky načte všechny dostupné lokalizace a zobrazí je v menu „*Nastavení – Jazyk*“.



**Upozornění:** *V případě chybějících nebo změněných identifikátorů v souboru s překlady dojde při výběru dané jazykové lokalizace k pádu programu. V takovém případě zkontrolujte strukturu lokalizačního souboru podle dostupné šablony.*

*Pokud nedojde k načtení přidané lokalizace, zkontrolujte, zda je název souboru vytvořen podle výše uvedených pravidel a zejména zda obsahuje platný kód pro lokalizaci.*



**Tip:** Pro správné zobrazení písmen s diakritikou je třeba je převést na jejich unicode kódy. Pro snadnější převod můžete použít skript `native2ascii.exe` obsažený v Java 2 Standard Edition Development Kit 6 (JDK).

## 6.2. Přidání podpory pro další databázové systémy

Jednou z výhod aplikace je, že je snadno rozšiřitelná o podporu dalších databází (které lze pak testovat). Tuto výhodu přináší využití JDBC ovladačů pro přístup k databázi. V následujících odstavcích budou popsány jednotlivé kroky, které je třeba vykonat při přidání nového ovladače pro připojení k databázi.

### 6.2.1. Stažení a vložení JDBC ovladače

Ovladač JDBC je obvykle zabalen jako archiv ZIP nebo JAR. Ovladače lze získat z mnoha zdrojů. Většina dodavatelů databázových systémů poskytuje alespoň jeden JDBC ovladač, který lze s jejich databázemi používat (obvykle bezplatně). Ovladače jsou prodávány i jinými dodavateli a nezdá se stává, že právě ovladače třetích stran poskytují vyšší výkon a spolehlivost než implementace poskytované dodavateli databáze. Seznam známých ovladačů je dostupný na adrese <http://developers.sun.com/product/jdbc/drivers>.

Pro snadné pochopení si ukážeme přidání JDBC ovladače k databázi MySQL. V prvním kroku je třeba stáhnout příslušný ovladač ze stránek společnosti MySQL AB. Ovladač pro MySQL 5.0 je dostupný na <http://dev.mysql.com/downloads/connector/j/5.0.html>. Stažený JDBC ovladač zabalený v archivu `tar.gz` rozbalte a samotný soubor `mysql-connector-java-5.0.7-bin.jar` nakopírujte do adresáře `/drivers/libs` v adresáři, kde máte uložen program `JSQLBenchmark`. Pokud chcete v seznamu JDBC ovladačů zobrazovat logo, vytvořte si obrázek o velikosti 151 x 80 pixelů, do kterého jej umístíte (většinou je možné ho získat ze stránek výrobce v různých velikostech). V žádném případě ale není logo povinně vyžadováno, v případě, že obrázek s logem nespecifikujete, použije aplikace defaultní (obrázek s bílým pozadím a černým rámečkem).

### 6.2.2. Nastavení parametrů JDBC ovladače

V dalším kroku je třeba nastavit základní údaje o JDBC ovladači, aby ho mohla aplikace načíst a zobrazit v seznamu dostupných ovladačů. Toto nastavení se provádí v souboru `/drivers/drivers_list.xml`. V tomto souboru má každý ovladač svůj záznam ohraničený elementy `<Driver></Driver>`, uvnitř kterých je uvedeno jméno, identifikátor, cesta k souboru s logem, stránky výrobce, příklad připojovacího řetězce (URL) a název třídy s JDBC ovladačem. Pro ovladač k databázi MySQL může tento záznam vypadat následovně:

```
<Driver id_number="1">
  <Name>MySQL</Name>
  <Website>http://www.mysql.com/</Website>
  <Logo>drivers/logo/logo_mysql.gif</Logo>
  <Identifier>mysql</Identifier>
  <UrlWizard>
    jdbc:mysql://&lt;server&gt;:&lt;port3306&gt;/&lt;database&gt;
  </UrlWizard>
  <DriverClass>com.mysql.jdbc.Driver</DriverClass>
</Driver>
```

Nejdůležitější položkou je jméno třídy JDBC ovladače (pokud ji neuvedete, nebude možné ovladač používat!). Ostatní údaje nejsou striktně vyžadovány, ale je vhodné je vyplnit, aby bylo použití ovladače v aplikaci snadnější (minimálně jméno a příklad URL adresy).



**Upozornění:** *Při nespecifikování parametru `DriverClass` nebude možné příslušný JDBC ovladač používat! Absence ostatních údajů nemá vliv na funkčnost nástroje, ale značně ztíží jeho použití.*

### 6.2.3. Kontrola načtení JDBC ovladače

Pokud jste celý postup provedli správně, měli byste po spuštění aplikace v seznamu databází vidět nově přidaný ovladač.

## 7. Klávesové zkratky

Pro rychlejší práci s programem je možné používat i některé předdefinované klávesové zkratky. Jejich přehled je uveden v tabulce 7.1.

Klávesová zkratka	Význam
Ctrl+O	Načtení výsledků testu ze souboru.
Ctrl+S	Uložení výsledků testu do souboru.
Shift+F3	Připojení k databázi.
Shift+F4	Odpojení od databáze.
Ctrl+N	Nový test.
Ctrl+Enter	Spustit test.

**Tabulka 7.1 Klávesové zkratky**